

**Vysoká škola báňská – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra telekomunikační techniky**

**Modelování obsluhových systémů pomocí nástroje Octave  
Queueing Package**

**The Queueing Systems Modelling Using the Octave Queueing  
Package**

**2014**

**Lukáš Kadlec**

## Zadání bakalářské práce

Student: **Lukáš Kadlec**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2601R013 Telekomunikační technika

Téma: Modelování obsluhových systémů pomocí nástroje Octave Queueing Package  
The Queueing Systems Modelling Using the Octave Queueing Package

Zásady pro vypracování:

1. Proved'te analýzu nástroje Octave Queueing Package
2. Vytvořte simulaci různých typů obsluhových systémů pomocí nástroje Octave Queueing Package
3. Celý postup zdokumentujte a vytvořte zadání laboratorního cvičení do odborného předmětu.

Seznam doporučené odborné literatury:

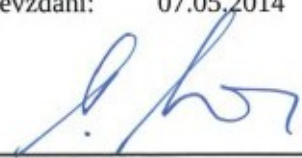
The Octave Queueing Package. MARZOLLA, Moreno. [online]. 2012 [cit. 2012-11-08]. Dostupné z: <http://www.moreno.marzolla.name/software/queueing/>  
GROSS, Donald. *Fundamentals of Queueing Theory*. 4th ed. John Wiley, 2009. ISBN 978-047-0547-830.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Libor Michalek, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

  
doc. Ing. Miroslav Vozňák, Ph.D.  
vedoucí katedry

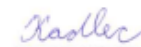


  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne: 5. května 2014



.....  
podpis studenta

## **Poděkování**

Rád bych poděkoval panu Ing. Liborovi Michalkovi, Ph.D., za odbornou pomoc a konzultaci při vytváření této bakalářské/diplomové práce.

## **Abstrakt**

Bakalářská práce popisuje teorii obsluhových systémů a zaměřuje se na jejich modelování v programu GNU Octave s balíčkem Queueing Package. V českém jazyce zatím není k dispozici žádný manuál pro tento balíček, a tak je práce určena pro základní práci s balíčkem s popsáním jednotlivých podporovaných funkcí. Práce by měla sloužit jako návod pro začátečníky a ukázat jim, jak postupovat při volání funkcí v Queueing Package. Do práce je zahrnuto i několik příkladů k daným obsluhovým systémům a Markovovým řetězcům. Pro vyzkoušení základních typů obsluhových systémů je v práci vytvořeno zadání laboratorního cvičení i s odpovídajícím řešením.

## **Klíčová slova**

Obsluhový systém, Markovovy řetězce, Octave, Queueing, Kendallova klasifikace, matice pravděpodobnosti přechodu, matice intenzit, režim fronty, Erlang, Engset, birth-death proces.

## **Abstract**

Bachelor thesis describes the theories of queueing systems, focusing on their modeling in the GNU Octave program with the Queueing Package. In the Czech language, yet there is no instruction manual for this package, so the work is designed for the elementary work with the package including a description of particular supported functions. The work should be used as a guide for beginners and show them how to proceed when calling functions in Queueing Package. The thesis also includes some examples of given queueing systems and Markov chains. For testing the basic types of service systems the thesis also includes a laboratory exercise with an adequate solution.

## **Key words**

Queueing system, Markov chains, Octave, Queueing, Kendall's notation, transition matrix, infinitesimal generator matrix, queueing discipline, Erlang, Engset, birth-death process.

## Seznam použitých zkratk a symbolů

OS	Obsluhový systém
FIFO	Režim fronty
LIFO	Režim fronty
GNU	Linuxový projekt
PCT1	Typ vstupního toku
PCT2	Typ vstupního toku
HPH	Hlavní provozní hodina
$n$	Počet obsluhových linek
Y	Obsloužené zatížení
A	Nabízené zatížení
R	Nezpracované zatížení
s	Počet účastníků
$c(0)$	Denní počet volání účastníka
$t(0)$	Střední doba obsazení spojovací cesty při jednom volání
k	Koncentrace
T	Pozorovací doba
Z	Ztráty
$C_z$	Počet ztracených požadavků
$C_n$	Počet všech požadavků
M	Poissonovský proces (exponenciální rozložení mezi příchody požadavků)
D	Deterministické rozložení
E	Erlangovo rozložení
H	Hyper-exponenciální rozložení
G	Jakékoliv rozložení
$\lambda$	Intenzita vstupního toku
$\mu$	Intenzita obsluhy
$L_i$	Očekávaný počet návštěv

# Obsah

<b>Úvod</b> .....	<b>10</b>
<b>1 Obsluhový systém</b> .....	<b>11</b>
1.1 Vstupní tok .....	12
1.1.1 Poissonovský tok.....	12
1.2 Režim fronty.....	13
1.3 Výstupní tok .....	13
<b>2 Provozní zatížení</b> .....	<b>14</b>
<b>3 Kendallova klasifikace</b> .....	<b>16</b>
<b>4 Modely OS</b> .....	<b>17</b>
4.1 Erlangův model .....	17
4.2 Engsetův model .....	17
<b>5 Markovovy řetězce</b> .....	<b>18</b>
5.1 Diskrétní Markovovy řetězce .....	18
5.2 Spojité Markovovy řetězce.....	18
<b>6 Program GNU Octave</b> .....	<b>20</b>
6.1 Octave Queueing Package.....	20
6.2 Instalace Octave Queueing Package.....	21
6.3 Konvence příkazů.....	21
<b>7 Modelování OS s jednou stanicí</b> .....	<b>24</b>
7.1 M/M/1 systém.....	24
7.2 M/M/m systém.....	28
7.3 Aplikace Erlang-B .....	29
7.4 Aplikace Erlang-C .....	30
7.5 Aplikace Engsetova vzorce .....	31
7.6 M/M/inf systém .....	32
7.7 M/M/1/K systém.....	33
<b>8 Modelování Markovových řetězců s diskrétním časem</b> .....	<b>36</b>
8.1 Pravděpodobnost obsazení stavů.....	36
8.2 Birth-Death proces.....	39



8.3 Očekávaný počet návštěv .....	39
8.4 Střední čas absorpce .....	41
8.5 Doba prvního přechodu .....	42
<b>9 Modelování Markovových řetězců se spojitým časem .....</b>	<b>44</b>
9.1 Pravděpodobnost obsazení stavů .....	44
9.2 Birth-Death proces .....	45
9.3 Střední čas absorpce .....	46
9.4 Doba prvního přechodu .....	46
<b>10 Návrh laboratorního cvičení .....</b>	<b>48</b>
<b>Závěr .....</b>	<b>50</b>

## Úvod

Bakalářská práce je rozdělena na teoretickou a praktickou část. Každá část obsahuje několik kapitol k dané problematice, které popisují základní popisy, pojmy a principy týkající se obsluhových systémů a jejich modelování v open-source programu GNU Octave. Obecný popis teorie obsluhových systémů je uveden ve druhé kapitole této práce. V bakalářské práci se zmiňuji o rozdělení obsluhových systémů, jejich klasifikaci, o provozním zatížení a jeho parametrech a popisuji jednotlivé části obsluhového systému. Teoretickou část zakončuji popisem Markovových řetězců s popisem základních typů OS.

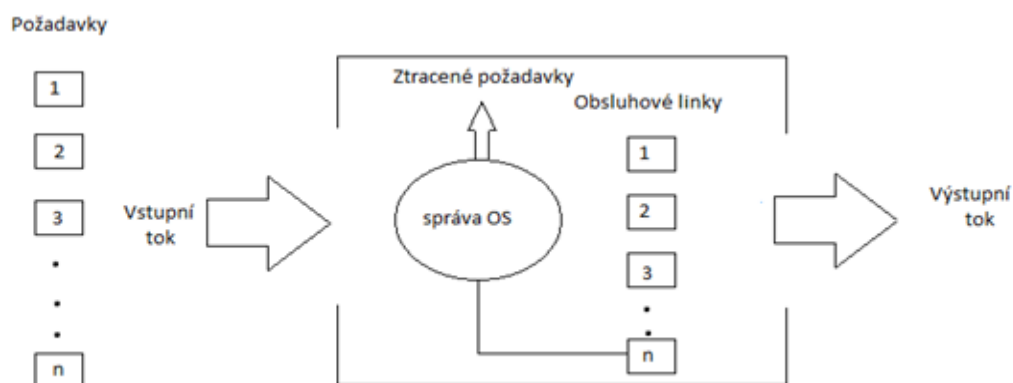
V praktické části jsem se věnoval popisu programu GNU Octave, instalaci toho programu a jeho rozšiřujícího balíčku Queueing Package pod operačním systémem Linux. Uvedl jsem základní konvenci příkazů v tomto programu s popisem jednotlivých funkcí. Následuje kapitola věnující se popisu jednotlivých obsluhových systémů, jejich vstupních a výstupních parametrů a také modelování těchto systémů. Modelování Markovových řetězců je další kapitola zahrnující popis jednotlivých funkcí, které je možné v programu GNU Octave provést. Poslední kapitolou mé práce je návrh laboratorního cvičení pro odborný předmět.

# 1 Obsluhový systém

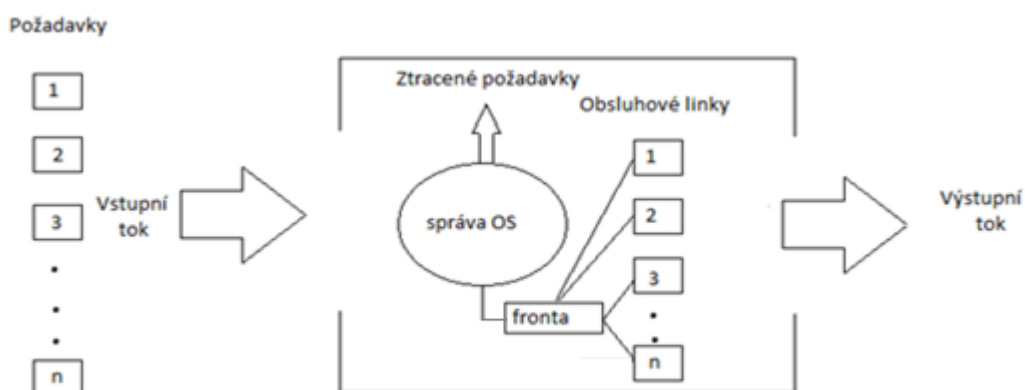
Obsluhovým systémem (dále jen OS) rozumíme topologické uspořádání jedné nebo několika obsluhových linek, které zpracovávají požadavky a dále definují, jak budou tyto požadavky obsluhovány. OS většinou umožňuje čekání na vyřízení požadavků viz **Obr. 1.2**, ale existují i systémy bez možnosti čekání viz **Obr. 1.1**. Mezi příklady obsluhových systémů můžeme z běžného života například uvést čekání ve frontě na lístek na vlak a ve frontě v lékárně nebo při volání operátora. U OS bez možnosti čekání dochází při obsazenosti všech obsluhových linek ke ztrátě. Každý obsluhový systém se skládá z několika faktorů:

N obsluhových linek, které vyřizují požadavky přicházející z S zdrojů (vstupní tok)

R míst určených k čekání pro vyřízení požadavků



Obrázek 1.1: OS bez možnosti čekání na obsluhu



Obrázek 1.2: OS s možností čekání na obsluhu

---

Zdroje, které generují požadavky poslané do OS, vytváří vstupní tok. Při vstupu požadavku do systému je přijat správcem OS, který určuje, zdali bude požadavek zpracován či nebude. Při přijetí se požadavek buď začne vyřizovat, nebo v případě obsazenosti obsluhových linek je řazen do fronty. Po vyřízení obsluhovou linkou je požadavek ukončen a vytváří další část OS, výstupní tok.

## 1.1 Vstupní tok

Definuje zdroj, ze kterého požadavky přicházejí a charakter požadavků. Můžeme rozdělit příchozí požadavky podle okamžiků příchodů do OS na náhodné (tzv. stochastické) nebo konstantní. Každý vstupní tok můžeme charakterizovat rozdělením dob mezi příchody:

**Rekurentní tok** - posloupnost nezávislých příchodů, které mají stejné rozdělení pravděpodobnosti

**Regenerativní tok** - posloupnost náhodných nezávislých příchodů

**Ordinární tok** - ve velmi krátkém časovém intervalu se pravděpodobnost příchodu dvou požadavků blíží nule

**Stacionární tok** - počet příchodů závisí na zvolené době časového intervalu

Modely OS systémů se od sebe liší i vstupem náhodného příchodu požadavků (vstupního toku). Nejčastěji se v praxi používají dva typy označované jako PCT1 a PCT2 (pure chance traffic).

PCT1 předpokládá, že vstupní požadavky přicházejí z velkého počtu zdrojů požadavků. Všechny tyto zdroje pak vytváří své požadavky jen zřídka, takže téměř neovlivňuje celkový počet požadavků. PCT2 naopak počítá s konečným počtem zdrojů, a tudíž i s nezanedbatelným vlivem jednoho zdroje na celkový vstupní tok[2].

### 1.1.1 Poissonovský tok

Nejpoužívanější model vstupního toku. Používaný tehdy, když množina vstupních požadavků (zdrojů) pochází z množiny vzájemně nezávislých uživatelů. Poissonův tok musí splňovat tři požadavky:

1. Počet příchodů je závislý na délce sledovaného intervalu.
2. Ve velmi krátkém časovém intervalu nenastanou dva požadavky.
3. Posloupnost náhodných příchodů je nezávislá.

Z toho vyplývá, že Poissonovský tok je regenerativní, rekurentní a ordinární.

Můžeme určit, jaká je pravděpodobnost vzniku  $k$  volání za čas  $t$ , podle **rovnice 1.1**:

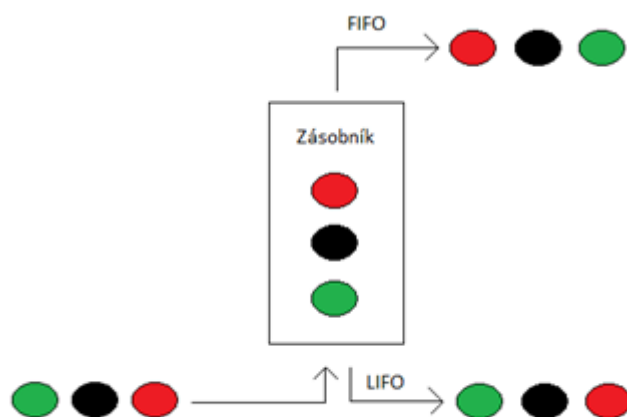
---


$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t} \quad (1.1)$$

## 1.2 Režim fronty

Režim fronty uplatníme v případě, že v okamžiku uvolnění obsluhové linky čeká ve frontě více než jeden požadavek. Každý požadavek, co se týče pořadí zpracování, je pak dále zpracován podle typu fronty. Nejpoužívanější režimy jsou:

- FIFO (First In First Out) - požadavek, který přijde do OS jako první, je také jako první obslužen viz **Obr. 1.3**. Někdy se používá termín řádný nebo přirozený režim.
- SIRO (náhodný režim) - Náhodné vybrání požadavku z fronty. V praxi se nepoužívá.
- LIFO (Last In First Out) - požadavek, který přijde do OS jako poslední, je obslužen jako první viz **Obr. 1.3**. Někdy též nazýván jako inverzní režim[4].
- PRI - podle priorit (po uvolnění kanálu obsluhy je vybírán požadavek s nejvyšší prioritou).



Obrázek 1.3: Fronta FIFO a LIFO

## 1.3 Výstupní tok

Je charakterizován ukončením obsluhy požadavků, které jsou vyřízeny v obsluhové lince. Výstupní tok je významný hlavně v případech, kdy tvoří vstupní tok do dalšího obsluhového systému[5].

---

## 2 Provozní zatížení

Základní kvantitativní parametr všech OS. Udává obsazenost všech obsluhových linek za určitou dobu  $T$ . Zdrojem zatížení je obecně vstupní požadavek (člověk, data, telefon). Můžeme ho určit vzhledem k jedné obsluhové lince nebo ke skupině linek.

Jednotkou byl ustanoven erlang [erl]. Každá obsluhová linka je schopna maximálně zpracovat intenzitu 1 erl, a to v případě, že je trvale obsazena po dobu jedné hodiny[2]. Provozní zatížení můžeme určit podle **rovnice 2.1**:

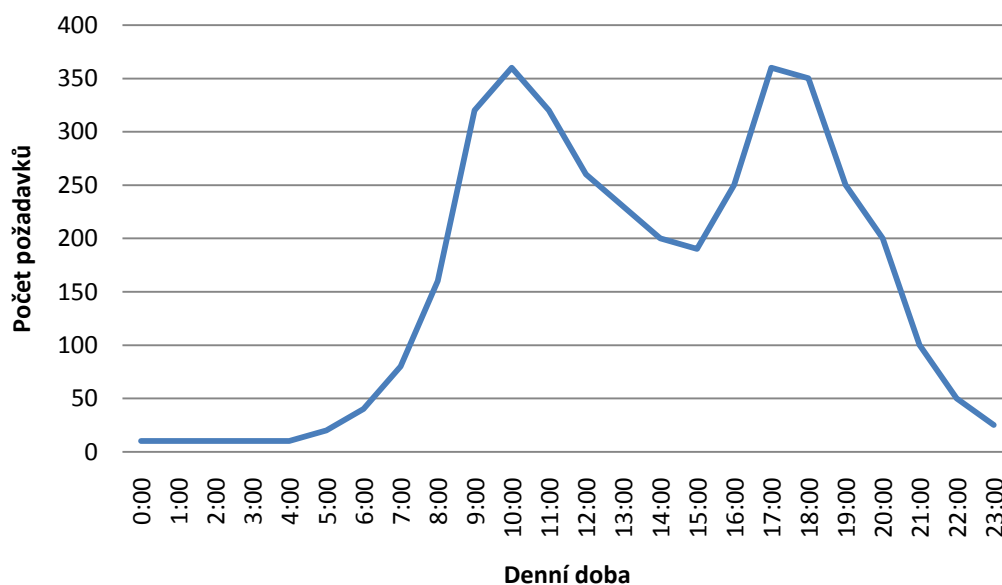
$$Y = \frac{\sum_{i=1}^n t_i}{T} \quad (2.1)$$

kde:

$n$  je počet obsluhových linek,  
 $t_i$  je doba obsazení linky,  
 $T$  udává pozorovací dobu.

Provozní zatížení se obvykle zjišťuje pozorováním nebo měřením. V průběhu dne existují dvě špičky provozu viz **Obr. 2.1**:

- Dopoludní
- Odpolední



Obrázek 2.1: Provozní špičky

---

Jako hlavní parametr provozního zatížení definujeme hlavní provozní hodinu (HPH), která udává čtyři po sobě jdoucí 15 minutové intervaly s nejvyšším provozním zatížením. HPH zahrnuje zhruba 15% z celkového zatížení v průběhu jednoho dne. Dalšími parametry pro provozní zatížení jsou:

- **AHT** (Average Hold Time) určuje průměrnou dobu čekání ve frontě.
- **CPS** (Call per Second) udává počet volání za sekundu.
- **BHCC** (Busy Hour Call Completions) udává hodinu během které je sestaveno nejvíce volání.
- **BHCA** (Busy Hour Call Attempts) udává hodinu s největším počtem volání.
- **BHT** (Busy Hour Traffic) je hlavní provozní hodina HPH.

Provozní zatížení můžeme rozdělit na:

**Obsloužené** - intenzita přenášeného zatížení

$$Y = \frac{\sum t_i}{60min} [erlang] \quad (2.2)$$

**Nabízené** - udává zatížení nabízené ústředně ke zpracování

$$A = \frac{\sum t_i}{60} [erlang] \quad (2.3)$$

**Nezpracované**

$$R = A - Y \quad (2.4)$$

Nabízené zatížení můžeme vypočítat i z následující **rovnice 2.5**:

$$A = s \cdot c(0) \cdot t(0) \cdot k \quad (2.5)$$

**s** je počet účastníků,  
**c(0)** je denní počet volání účastníka,  
**t(0)** je střední doba obsazení obsluhové linky při jednom volání,  
**k** je koncentrace.

U obsluhových systémů bez možnosti zařazení požadavků do front, můžeme určit ztráty **Z** podle **rovnice 2.6**:

$$Z = \frac{C_z}{C_n} \quad (2.6)$$

**C<sub>z</sub>** je počet ztracených požadavků,  
**C<sub>n</sub>** je počet všech požadavků[2].

---

### 3 Kendallova klasifikace

Popisuje zákonitosti vzniku a příchodu požadavku do OS, dále jak jsou požadavky zpracovány v případě obsazenosti obsluhových linek pomocí front a udává počet obsluhových linek. Každý systém je tříděn podle tří kritérií[1]:

1. podle typu náhodného procesu, který popisuje vstupní tok do OS.
2. podle zákona rozložení délky obsluhy.
3. podle množství obsluhových linek, které zpracovávají požadavky.

Každý systém je zapsán ve tvaru  $A/B/n$  podle zkrácené verze nebo ve tvaru  $A/B/N/Y/S/X$  dle rozšířené verze Kendallovy klasifikace.

Význam jednotlivých symbolů:

**A** typ vstupního toku požadavků do systému, každý vstupní tok nese další jednotlivé označení:

- M** Poissonovský proces (exponenciální rozložení mezi příchody)
- D** Deterministické rozložení
- E** Erlangovo rozložení
- H** Hyper-exponenciální rozložení
- G** Jakékoliv rozložení

**B** ... zákon rozložení délky trvání obsluhy požadavku

**N** ... udává množství obsluhových linek, které zpracovávají požadavky

**Y** ... nejvyšší možný počet požadavků vstupujících do systému

**S** ... počet zdrojů požadavků, neuvádí se, pokud je nekonečný

**X** ... udává, v jakém režimu fronta zpracovává přijaté požadavky



---

## 4 Modely OS

### 4.1 Erlangův model

Model, který předpokládá vstupní tok PCT1 s konečným počtem vedení (např. M/M/N/0). Tento model je nejvíce využíván v teorii i v početní části telegrafních a telefonních sítí a je též označován jako Erlangův ztrátový model.

Pro intenzitu vstupního toku platí:  $\lambda_j = \lambda$  pro  $j = 0, 1, 2, 3, \dots, N$  a pro intenzitu obsluhy:  $\mu_j = j \mu$ .

Pro popis základních vztahů mezi parametry se využívá takzvaný Erlangův ztrátový vzorec nebo Erlangův první vzorec (vztah) označovaný  $E_N(A)$  nebo  $E_{1,N}(A)$ [2] viz **rovnice 4.2**.

$$P_N = E_{1,N}(A) = \frac{\frac{A^N}{N!}}{\sum_{i=0}^N \frac{A^i}{i!}} \quad (4.1)$$

Rovnice uvádí, s jakou pravděpodobností bude počet kanálů  $N$  obsazených. V případě, že známe nabízené zatížení  $A$  a  $N$  udává počet obsluhových kanálů, můžeme spočítat pravděpodobnost blokace neboli pravděpodobnost, že následující přicházející požadavek nebude obslužen a dojde ke ztrátě požadavku[1].

Pro určení pravděpodobnosti čekání požadavku ve frontě slouží Erlangův čekací vzorec nebo také Erlangův druhý vzorec viz **rovnice 4.2**, který se obecně označuje  $E_{2,N}(A)$ [2].

$$P(W > 0) = E_{2,N}(A) = \frac{\frac{A^N}{N!} * \frac{N}{N-A}}{\sum_{k=0}^{N-1} \frac{A^k}{k!} + \frac{A^N}{N!} * \frac{N}{N-A}} \quad (4.2)$$

### 4.2 Engsetův model

Model, který předpokládá konečný počet zdrojů a vstupní tok PCT 2[1]. Počet zdrojů je však mnohonásobně větší než počet vedení (M(n)/M/M/0). Engsetův model se používá pro pobočkové ústředny PBX[2].

Pro určení pravděpodobností počtu současně obsazených linek se používá Engsetova **rovnice 4.3**:

$$P_k = \frac{\binom{S}{k} \beta^k}{\sum_{i=0}^N \binom{S}{i} \beta^i} \quad (4.3)$$

---

## 5 Markovovy řetězce

### 5.1 Diskrétní Markovovy řetězce

Nechť  $X_0, X_1, \dots, X_n$  je posloupnost náhodných proměnných definovaných nad množinou diskrétních stavů  $1, 2, \dots$ . Posloupnost  $X_0, X_1, \dots, X_n$  je stochastický proces s diskrétním časem  $0, 1, 2, \dots$ . Markovův řetězec je stochastický proces  $\{X_n, n = 0, 1, 2, \dots\}$ , který splňuje následující Markovovu vlastnost:

$$P(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, X_0 = x_0) = P(X_{n+1} = x_{n+1} | X_n = x_n),$$

což v podstatě znamená, že pravděpodobnost toho, že systém je v konkrétním stavu v čase  $n+1$ , závisí pouze na tom, v jakém stavu byl systém v čase  $n$ .

Vývoj Markovova řetězce s konečnou množinou stavů  $\{1, 2, \dots, N\}$  může být popsán stochastickou maticí  $P(n) = [P_{i,j}(n)]$ , což znamená  $P_{i,j}(n) = P(X_{n+1} = j | X_n = i)$ . Pokud je Markovův řetězec homogenní (matice přechodu  $P(n)$  je časově nezávislá), můžeme psát  $P = [P_{i,j}]$ , kde  $P_{i,j} = P(X_{n+1} = j | X_n = i)$  pro všechna  $n = 0, 1, \dots$  [9].

Matice přechodu  $P$  musí splňovat následující dvě vlastnosti:

- 1)  $P_{i,j} \geq 0$  pro všechna  $i, j$
- 2)  $\sum_{j=1}^N P_{i,j} = 1$  pro všechna  $i$ .

### 5.2 Spojité Markovovy řetězce

Stochastický proces  $\{X(t), t \geq 0\}$  je Markovův řetězec se spojitým časem tehdy, když pro všechna čísla  $n$  a pro nějaké sekvence  $t_0, t_1, \dots, t_n, t_{n+1}$ , tak že  $t_0 < t_1 < t_n < t_{n+1}$ , máme:

$$\begin{aligned} P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n, X(t_{n-1}) = x_{n-1}, \dots, X(t_0) = x_0) \\ = P(X(t_{n+1}) = x_{n+1} | X(t_n) = x_n) \end{aligned}$$

Markovův řetězec se spojitým časem je definován v závislosti na matici intenzit  $Q = [Q_{i,j}]$ , kde pro každé  $i \neq j$ ,  $Q_{i,j}$  je hodnota přechodu ze stavu  $i$  do stavu  $j$ . Matice  $Q$  musí splňovat podmínku, že pro všechna  $i$  platí:  $\sum_{j=1}^N Q_{i,j} = 0$  [8].

---

Pravděpodobnost přechodu mezi dvěma stavy  $e_i$  a  $e_j$  označíme  $p_{ij}$ . Určíme, s jakou pravděpodobností se systém po dvou dalších krocích nachází ve stavu  $e_j$ . Při prvním kroku přejde systém ze stavu  $e_i$  do nového stavu  $e_k$ . Pravděpodobnost tohoto přechodu označme  $p_{ik}$ . V dalším kroku přejde systém z  $e_k$  do  $e_j$ , poněvadž pravděpodobnost tohoto přechodu není nijak ovlivněna předchozími kroky. Pravděpodobnost druhého přechodu označme  $p_{kj}$ .

Pro pravděpodobnost  $p_{ij}$  tak podle pravidla o sčítání pravděpodobností (**rovnice 5.1**) platí[7]:

$$p_{ij} = \sum_k p_{ik} p_{kj} \quad (5.1)$$

Obecnější vzorec pro větší počet přechodů po  $n$  krocích (**rovnice 5.2**):

$$p_{ij} = \sum_k p_{ik}^{(n-1)} p_{kj} \quad (5.2)$$

---

## 6 Program GNU Octave

Program Octave je volně šiřitelný (tzv. open-source) matematický program vytvořený pod projektem GNU podobný komerčnímu programu Matlab. GNU projekt byl vytvořen v roce 1983 za účelem vytvoření svobodného operačního systému. Octave byl určen pro matematické výpočty, ačkoliv dnes existuje řada grafických nadstaveb. Program lze v modifikovaných verzích nalézt na různých platformách (Unix, MS Windows, Mac OS)[3].

### 6.1 OctaveQueueingPackage

Doplňující balíček Queueing, dříve známý jako qnetworsk, je kolekce funkcí napsaných v GNU Octave pro analýzu obsluhových systémů a Markovových řetězců. Konkrétně balíček Queueing obsahuje funkce pro analýzu Jacksonových sítí, otevřených, uzavřených nebo smíšených BCMP sítí a výpočet mezí výkonu. Pro výpočty byly použity následující algoritmy:

- Přesné a přibližné analýzy střední hodnoty pro uzavřené síť;
- Analýza střední hodnoty pro smíšené síť;
- Analýza střední hodnoty pro síť s blokováním;
- Asymptotické meze, symetrický systém mezí a geometrické meze;

Queueing poskytuje funkce pro analýzu následujících typů systémů hromadné obsluhy:

- M/M/1
- M/M/m
- M/M/ $\infty$
- M/M/1/k (1 obsluhová linka, konečný počet požadavků)
- M/M/m/k (více obsluhových linek, konečný počet požadavků)
- Asymetrický M/M/1
- M/G/1
- M/H/13

Pro analýzu Markovových řetězců jsou k dispozici funkce:

- Birth-Death procesy
- Průměrná doba absorpce
- Střední doba prvního přechodu
- Očekávaná doba pobytu

---

## 6.2 Instalace OctaveQueueingPackage

Pokud používáte současnou verzi Programu GNU Octave, můžete nainstalovat QueueingPackage přímo z příkazového řádku programu Octave:

```
pkg install - forge queueing
```

Výše uvedený příkaz automaticky stáhne a nainstaluje nejnovější verzi balíčku Queueing z OctaveForge do vašeho počítače. Při problémech s oprávněním pro zápis do instalační složky, použijte příkaz `local`, který nainstaluje balíček do domovské složky:

```
pkg install -forge -local queueing
```

Ověřte si, zda byl balíček nainstalován pomocí příkazu:

```
pkg list queueing
```

Od verze 1.1.1 se balíček automaticky nenačítá při spuštění Octave. Aby byl k dispozici k použití, použijte následující příkaz:

```
pkg load queueing
```

## 6.3 Konvence příkazů

Většina příkazů v balíčku Queueing má společné pojmenování funkcí. Názvy funkcí se skládají z několika částí, první z nich je prefix, který označuje třídu problémů funkčních adres:

- ctmc*- funkce pro Markovovy řetězce se spojitým časem
- dtmc*- funkce pro Markovovy řetězce s diskrétním časem
- qs*- funkce pro analýzu obecných systémů hromadné obsluhy

---

*qn-* funkce pro analýzu sítí hromadné obsluhy

Funkce, které se zabývají Markovovými řetězci, začínají prefixem buď *ctmc* nebo *dtmc*. Prefix je volitelně následován dalším řetězcem, který určuje, co funkce dělá:

<i>bd</i>	Birth-Death proces
<i>mtta</i>	Střední doba do absorpce
<i>fpt</i>	Doba prvního přechodu
<i>exps</i>	Předpokládaná doba pobytu
<i>taexps</i>	Průměrná očekávaná doba pobytu

Například funkce *ctmc bd* vrací pravděpodobnost matice přechodu pro spojitý Birth-Death procesy, zatímco *dtmc bd* vrací pravděpodobnost matice přechodu pro diskrétní Birth-Death procesy. Funkce začínající prefixem *qs* popisují systémy hromadné obsluhy, následující sufix popisuje typ systému, například *qsmm1* pro M/M/1 systém.

Funkce, jejichž název začíná na *qn-*, jsou spojeny se sítěmi obsluhových systémů. Znak, který následuje, udává, zda je síť otevřená ("o") nebo uzavřená ("c") a zda se jedná o jednu třídu klienta nebo o více tříd. Řetězec *mix* označuje, že funkce podporuje smíšené síť.

<i>-os-</i>	Otevřená síť s jednou třídou zákazníků
<i>-om-</i>	Otevřená síť s více třídami zákazníků
<i>-cs-</i>	Uzavřená síť s jednou třídou zákazníků
<i>-cm-</i>	Uzavřená síť s více třídami zákazníků
<i>-mix-</i>	Smíšená síť

Poslední část názvu funkce označuje algoritmy implementované funkce:

<i>-aba</i>	Analýza asymptotických mezí
<i>-gb</i>	Geometrické meze
<i>-cb</i>	Složené meze
<i>-mva</i>	Algoritmus analýzy střední hodnoty

---

<code>-cmva</code>	Podmíněná analýza střední hodnoty
<code>-mvaap</code>	Přibližná analýza střední hodnoty
<code>-conv</code>	Konvoluční algoritmus

Můžete použít všechny funkce pouhým vyvoláním jejich jména s příslušnými parametry, balíček Queueing by měl sám zobrazovat chybná hlášení v případě chybějících nebo špatně zadaných parametrů. K dispozici je rozsáhlá dokumentace pro každou funkci a lze ji zobrazit pomocí příkazu *help*. Například příkaz:

```
help qncsmvablo
```

vypíše dokumentaci pro funkci *qncsmvablo*. Další informace lze nalézt v příručce balíčku Queueing[8].

---

## 7 Modelování OS s jednou stanicí

### 7.1 M/M/1 systém

M/M/1 systém je složen z jednoho serveru připojeného k neomezenému počtu FIFO front. Jedná se o exponenciální rozdělení mezi příchody požadavků (Poissonův proces) s intenzitou příchodů žádostí  $\lambda$ . Doba obsluhy je exponenciální s průměrnou intenzitou obsluhy  $\mu$ . Systém je stabilní, pokud  $\lambda < \mu$ .

```
[ U, R, Q, X, p0 ] = qsmm1 (lambda, mu)
```

Vypočítá využití systému, čas strávený v systému, průměrný počet požadavků, propustnost a pravděpodobnost, že v systému nejsou žádné požadavky pro M/M/1 systém.

#### VSTUPNÍ PARAMETRY:

- *lambda* intenzita požadavků ( $\lambda \geq 0$ ).
- *mu* intenzita obsluhy ( $\mu > \lambda$ ).

#### VÝSTUPNÍ PARAMETRY:

- *U* Využití systému
- *R* Čas strávený v systému
- *Q* Průměrný počet požadavků v systému
- *X* Propustnost systému, je-li systém ergodický ( $\mu > \lambda$ ), vždy se  $X = \lambda$ .
- *p0* Ustálená pravděpodobnost, že v systému nejsou žádné požadavky

$\lambda$  a  $\mu$  mohou být také vektory o stejné velikosti. V tomto případě pak budou výsledkem též vektory[3].



---

Příklad M/M/1:

Máme zadaný M/M/1 systém s intenzitou příchoďů  $\lambda = 0,3$  požadavků / minutu a intenzitou obsluhy  $\mu = 0,4$  požadavků / minutu.

Určíme využití systému  $U$ , čas, který stráví požadavek v systému  $R$ , průměrný počet požadavků v systému  $Q$  a pravděpodobnost, že v systému nejsou žádné požadavky  $p_0$ .

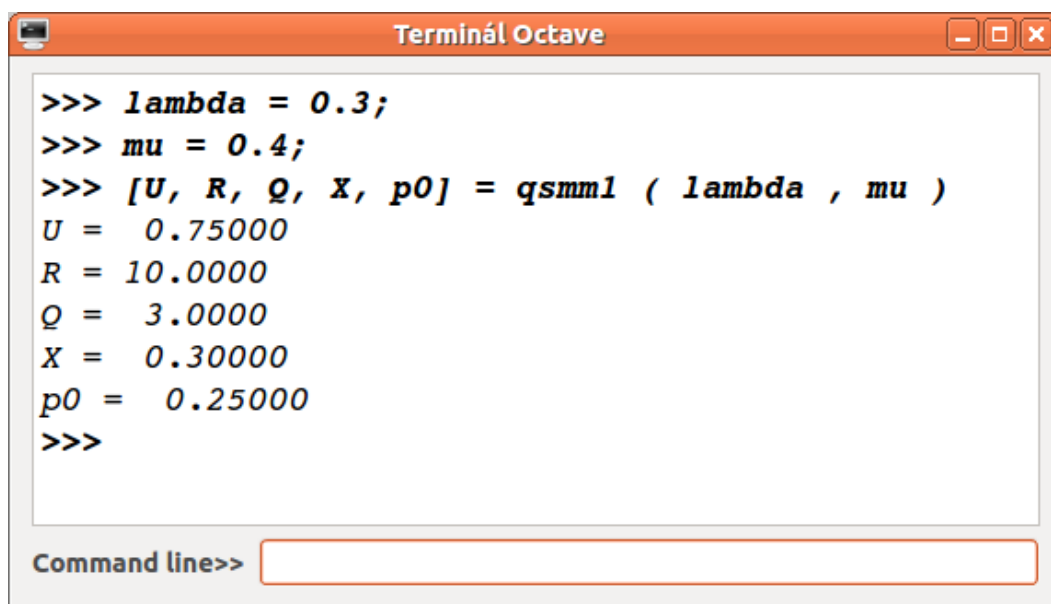
$$U = \frac{\lambda}{\mu} = \frac{0,3}{0,4} = 75\%$$

$$R = \frac{1}{\mu - \lambda} = 10 \text{ minut}$$

$$Q = \frac{\lambda}{\mu - \lambda} = 3 \text{ požadavky}$$

$$p(0) = 1 - \frac{\lambda}{\mu} = 25\%$$

Ukázka v Octave viz **Obr. 7.1**:



```
Terminál Octave

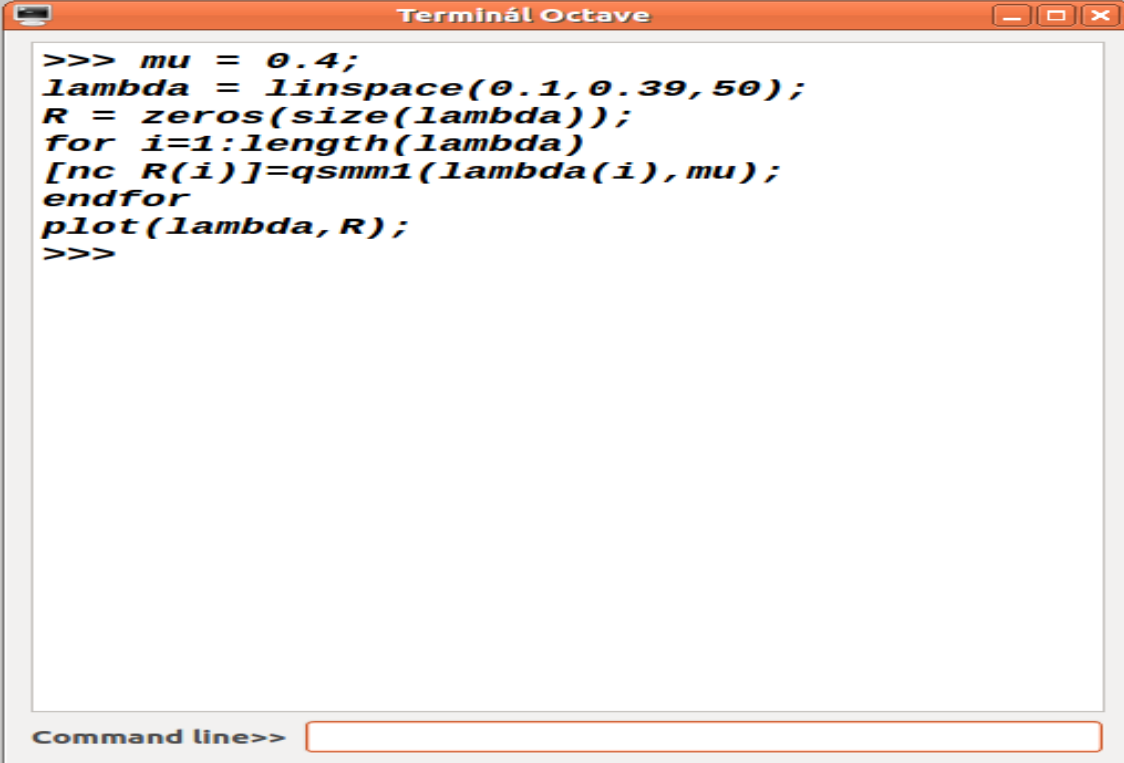
>>> lambda = 0.3;
>>> mu = 0.4;
>>> [U, R, Q, X, p0] = qsmml ( lambda , mu )
U = 0.75000
R = 10.0000
Q = 3.0000
X = 0.30000
p0 = 0.25000
>>>

Command line>> 
```

Obrázek 7.1: Modelování M/M/1 systému

---

Můžeme určit, jak čas strávený v systému  $R$  roste s příchodem požadavků s intenzitou obsluhy  $\mu = 0,4$  viz **Obr. 7.2**.



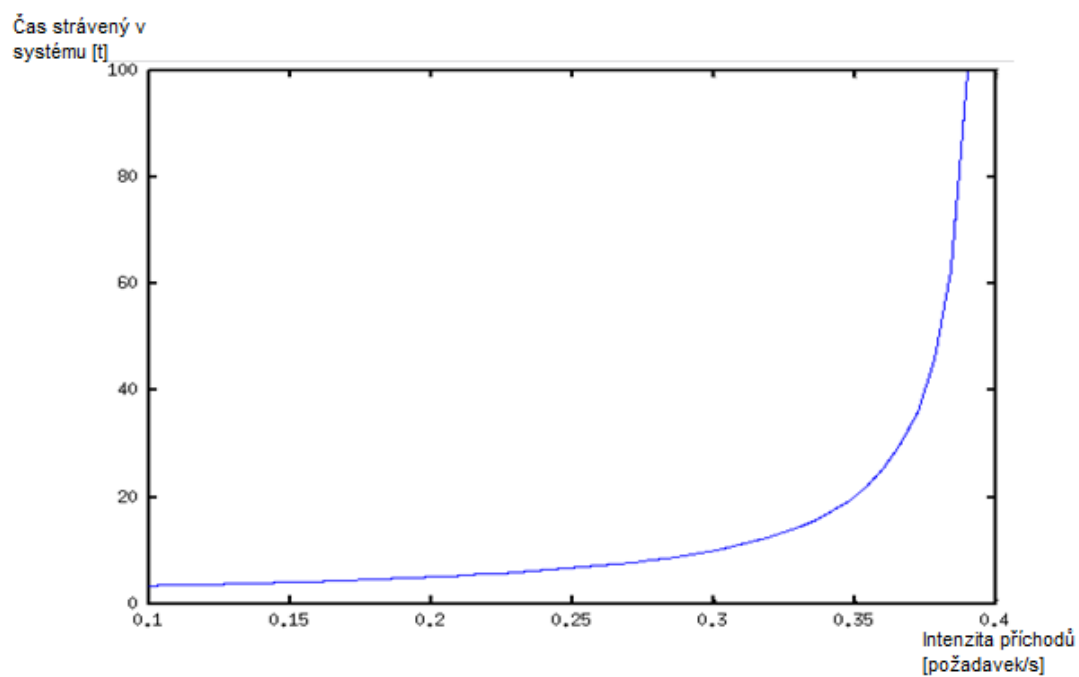
```
>>> mu = 0.4;
lambda = linspace(0.1, 0.39, 50);
R = zeros(size(lambda));
for i=1:length(lambda)
    [nc R(i)] = qsmm1(lambda(i), mu);
endfor
plot(lambda, R);
>>>
```

Command line>>

Obrázek 7.2: Kód pro vykreslení závislosti  $R$  na intenzitě příchoďů

---

Závislost času stráveného v systému na intenzitě příchodů **Obr. 7.3:**



Obrázek 7.3: Závislost času stráveného v systému na intenzitě příchodů

---

## 7.2 M/M/m systém

M/M/m systém je podobný M/M/1 systému kromě toho, že jsou zde identické servery  $m \geq 1$  připojené ke sdílené FIFO frontě. To znamená, že většina požadavků může být přijata ve stejný čas.

$$[U, R, Q, X, p_0, p_m] = \text{qsmmm}(\lambda, \mu)$$

$$[U, R, Q, X, p_0, p_m] = \text{qsmmm}(\lambda, \mu, m)$$

Vypočítá využití systému, čas strávený v systému, průměrný počet požadavků a propustnost pro M/M/m systém, kde je  $m$  obsluhových linek připojených ke sdílené FIFO frontě.

### VSTUPNÍ PARAMETRY:

- $\lambda$  intenzita příchodů ( $\lambda > 0$ ).
- $\mu$  intenzita obsluhy ( $\mu > \lambda$ ).
- $m$  počet serverů ( $m \geq 1$ ). Pokud je parametr vynechán, automaticky  $m=1$ .

### VÝSTUPNÍ PARAMETRY:

- $U$  Využití systému
- $R$  Čas strávený v systému
- $Q$  Průměrný počet požadavků v systému
- $X$  Propustnost systému, je-li systém ergodický ( $\mu > \lambda$ ), vždy se  $x = \lambda$ .
- $p_0$  Ustálená pravděpodobnost, že v systému nejsou žádné požadavky
- $p_m$  Ustálená pravděpodobnost, že přicházející žádost musí čekat ve frontě.

$\lambda$ ,  $\mu$  a  $m$  mohou být vektory stejné velikosti, výsledkem jsou opět vektory[3].

---

## 7.3 Aplikace Erlang-B

Vypočte hodnotu Erlang-B vzorce  $E_B(A, m)$  určující pravděpodobnost, že otevřený systém s  $m$  identickými servery, intenzitou příchodu  $\lambda$ , intenzitou obsluhy  $\mu$  a nabízeným zatížením  $A = \lambda / \mu$  má všechny servery obsazené.  $E_B(A, m)$  můžeme určit podle **rovnice 7.1**:

$$E_B(A, m) = \frac{A^m}{m!} \left( \sum_{k=0}^m \frac{A^k}{k!} \right)^{-1} \quad (7.1)$$

B = erlangb (A, m)

### VSTUPNÍ PARAMETRY:

- $A$  Nabízené zatížení, definované jako  $A = \lambda / \mu$ , kde  $\lambda$  je intenzita příchodů a  $\mu$  je intenzita obsluhy každého serveru ( $A > 0$ ).
- $m$  Počet identických serverů ( $m \geq 1$ ). Standardně  $m = 1$ .

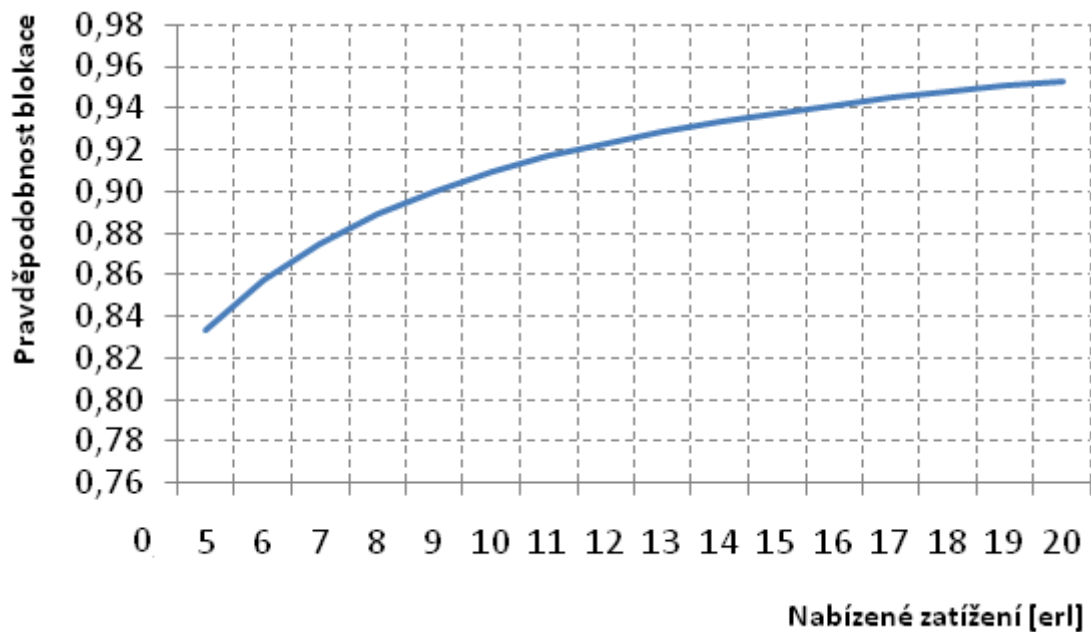
### VÝSTUPNÍ PARAMETRY:

- $B$  Hodnota  $E_B(A, m)$

$A$  nebo  $m$  mohou být vektory, v tom případě jsou výsledkem znovu vektory[3].

Příklad:

Máme zadáno nabízené zatížení  $A = 5, 6, 7, \dots, 20$ . Systém pracuje s jednou obsluhovou linkou. Určíme pravděpodobnost blokace požadavku pro každé nabízené zatížení a sestrojíme graf závislosti pravděpodobnosti blokace na nabízeném zatížení.



Obrázek 7.4: Závislost pravděpodobnosti blokace na nabízeném zatížení

## 7.4 Aplikace Erlang-C

Vypočte pravděpodobnost čekání  $E_C(A, m)$  pro otevřený systém s  $m$  identickými servery, nekonečnou frontou, intenzitou příchodu  $\lambda$ , intenzitou obsluhy  $\mu$  a nabízeným zatížením  $A = \lambda / \mu$ .  $E_C(A, m)$  je definováno jako:

$$E_C(A, m) = \frac{A^m}{m!} \frac{1}{1-p} \left( \sum_{k=0}^{m-1} \frac{A^k}{k!} + \frac{A^m}{M!} \frac{1}{1-p} \right)^{-1} \quad (7.2)$$

kde je  $p = A / m = \lambda / (m\mu)$ .

C = erlangc (A, m)

### VSTUPNÍ PARAMETRY:

- $A$  Nabízené zatížení, definované jako  $A = \lambda / \mu$ , kde  $\lambda$  je intenzita příchodů a  $\mu$  je intenzitou obsluhy každého serveru ( $0 < A < m$ ).
- $m$  Počet identických serverů ( $m \geq 1$ ). Standardně  $m = 1$ .

---

#### VÝSTUPNÍ PARAMETRY:

- $C$  Hodnota  $E_C(A, m)$

$A$  nebo  $m$  mohou být vektory, v tom případě výsledek budou zase vektory[3].

### 7.5 Aplikace Engsetova vzorce

Vypočítá Engsetovu pravděpodobnost blokování  $P_B(A, m, n)$  pro systém s konečným počtem  $n$  uživatelů,  $m$  identickými servery, žádnou frontou, intenzitou obsluhy  $\mu$ , intenzitou příchoďů  $\lambda$  a nabízeným zatížením  $A = \lambda / \mu$ .  $P_B(A, m, n)$  je definováno pro  $n > m$  a můžeme ho vypočítat podle **rovnice 7.3**:

$$P_B(A, m, n) = \frac{A^m \binom{n}{m}}{\sum_{k=0}^m A^k \binom{n}{k}} \quad (7.3)$$

$P_B$  je rovno 0, když  $m \geq n$ .

$B = \text{engset}(A, m, n)$

#### VSTUPNÍ PARAMETRY:

- $A$  Nabízené zatížení, definované jako  $A = \lambda / \mu$ , kde  $\lambda$  je střední intenzita příchoďů a  $\mu$  střední intenzitou obsluhy každého individuálního serveru ( $A > 0$ ).
- $m$  Počet identických serverů ( $m \geq 1$ ). Standardně  $m = 1$ .
- $n$  Počet požadavků ( $n \geq 1$ ). Standardně  $n = 1$ .

#### VÝSTUPNÍ PARAMETRY:

- $B$  Hodnota  $P_B(A, m, n)$

$A$ ,  $m$  nebo  $n$  mohou být vektory, v tom případě výsledkem budou zase vektory.

---

## 7.6 M/M/inf systém

M/M/∞ systém je podobný M/M/m systému kromě toho, že zde je nekonečně mnoho identických obsluhových linek (to znamená, že  $m = \infty$ ). Každý nový požadavek je přidělen jiné obsluhové lince, takže nikdo nemůže dojít k frontě. M/M/∞ je vždycky stabilní.

```
[ U, R, Q, X, p0] = qsmminf (lambda, mu)
```

Vypočítá využití, čas strávený v systému, průměrný počet požadavků a propustnost pro M/M/∞ systém. M/M/∞ systém má nekonečný počet identických obsluhových linek; tento typ systému je vždycky stabilní pro každou intenzitu příchodů a obsluhy.

Ustálená pravděpodobnost  $\pi_k$ , že v systému je  $k$ -požadavků,  $k \geq 0$ , může být spočítána podle rovnice 7.4:

$$\pi_k = \frac{1}{k!} \left( \frac{\lambda}{\mu} \right)^k e^{-\lambda / \mu} \quad (7.4)$$

### VSTUPNÍ PARAMETRY:

- *lambda* Intenzita příchodů ( $lambda > 0$ ).
- *mu* Intenzita obsluhy ( $mu > 0$ ).

### VÝSTUPNÍ PARAMETRY:

- *U* Intenzita provozu (definovaná jakou  $\lambda / \mu$ ). Všimněte si, že toho je odlišné od využití obsluhové linky, které je v tomto případě M/M/∞ vždy rovno nule.
- *R* Čas strávený v systému
- *Q* Průměrný počet požadavků v systému (je roven intenzitě provozu  $\lambda / \mu$ ).
- *X* Propustnost (je vždy rovna  $X = lambda$ ).
- *p0* Ustálená pravděpodobnost, že v systému nejsou žádné požadavky.

*lambda* a *mu* mohou být vektory stejné velikosti, v tom případě jsou výsledkem zase vektory.



---

## 7.7 M/M/1/K systém

V M/M/1/K systému s konečnou kapacitou je pouze jedna obsluhová linka a může zde být  $k \geq I$  požadavků (včetně již obsluhovaného požadavku). Když se nový požadavek snaží vstoupit do systému v době, když je zde  $K$  jiných požadavků, vstupující požadavek je odmítnut.

`[U, R, Q, X, p0, pK] = qsmmlk (lambda, mu, K)`

Vypočítá využití, čas strávený v systému, průměrný počet požadavků a propustnost pro M/M/1/K systém s konečnou kapacitou. V M/M/1/K systému je pouze jedna obsluhová linka; maximální počet požadavků je  $K$  a maximální délka fronty je  $K - I$ .

Ustálená pravděpodobnost  $\pi_k$ , že v systému je  $k$ -požadavků,  $0 \leq k \leq K$ , může být spočítána podle **rovnice 7.5**:

$$\pi_k = \frac{(1-a)a^k}{1-a^{K+1}} \quad (7.5)$$

kde  $a = \lambda / \mu$ .

### VSTUPNÍ PARAMETRY:

- *lambda*      Intenzita příchodů ( $lambda > 0$ ).
- *mu*          Intenzita obsluhy ( $mu > 0$ ).
- *K*            Maximální povolený počet požadavků v systému ( $K \geq I$ ).

### VÝSTUPNÍ PARAMETRY:

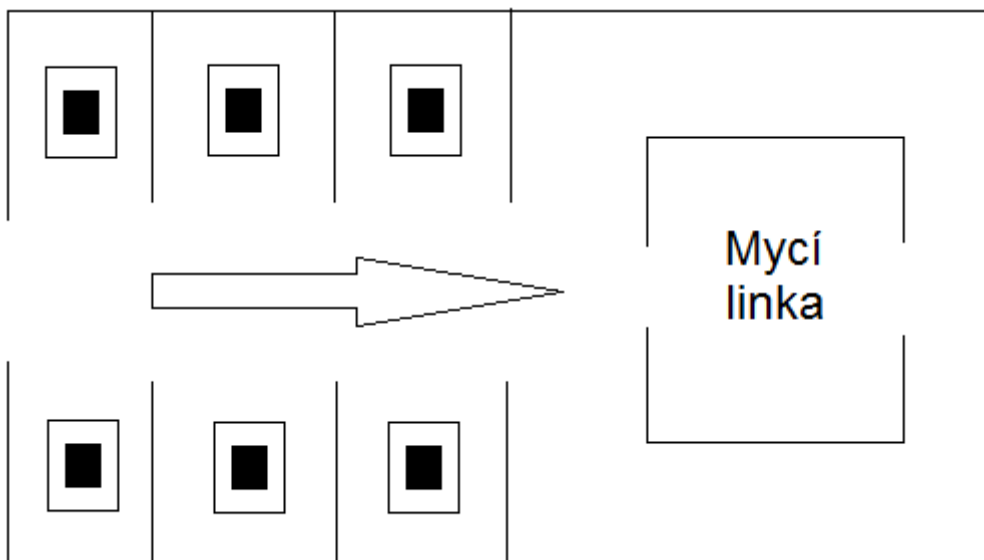
- *U*            Využití systému, které je definováno jako:  $U = I-p0$ .
- *R*            Čas strávený v systému
- *Q*            Průměrný počet požadavků v systému
- *X*            Propustnost systému
- *p0*          Ustálená pravděpodobnost, že v systému nejsou žádné požadavky.
- *pK*          Ustálená pravděpodobnost, že v systému je  $K$  požadavků.

*lambda*, *mu* a *K* mohou být vektory stejné velikosti, v tom případě výsledkem je zase vektor.

---

Příklad M/M/1/K systému:

Mějme mycí linku, před kterou je možné zaparkovat 6 aut do fronty. Mycí linka může obsluhovat pouze jedno auto. Auto přijíždí do myčky průměrně 2 minuty a doba obsluhy mytí je 6 minut. Průměrně do myčky přijede 9 aut za hodinu. Situaci vidíme na následujícím obrázku:



Obrázek 7.5: Příklad M/M/1/K systému

Intenzita příjezdů k myčce:

$$\lambda = 9 \text{ aut/hod.}$$

Označme  $T_s$  jako dobu mytí a  $T_j$  jako dobu, kterou auto vjíždí do myčky. Pak můžeme vypočítat dobu obsluhy  $T$ :

$$T = T_s + T_j = 2 + 6 = 8 \text{ minut}$$

Nyní vypočítáme intenzitu obsluhy následovně:

$$\mu = \frac{1}{T} = 7,5 \text{ aut/hod.}$$

Pravděpodobnost, že v systému nejsou žádné požadavky:

$$p(0) = \frac{1 - p}{1 - p^{k+1}} \approx 6\%$$

$$\text{kde } p = \frac{\lambda}{\mu}.$$

---

Využití systému:

$$U = 1 - p(0) = 94\%$$

Označme maximální kapacitu aut před myčkou  $Ws$  a maximální počet aut v myčce  $Wt$ , pak můžeme určit maximální počet požadavků v systému:

$$K = Ws + Wt = 6 + 1 = 7 \text{ požadavků}$$

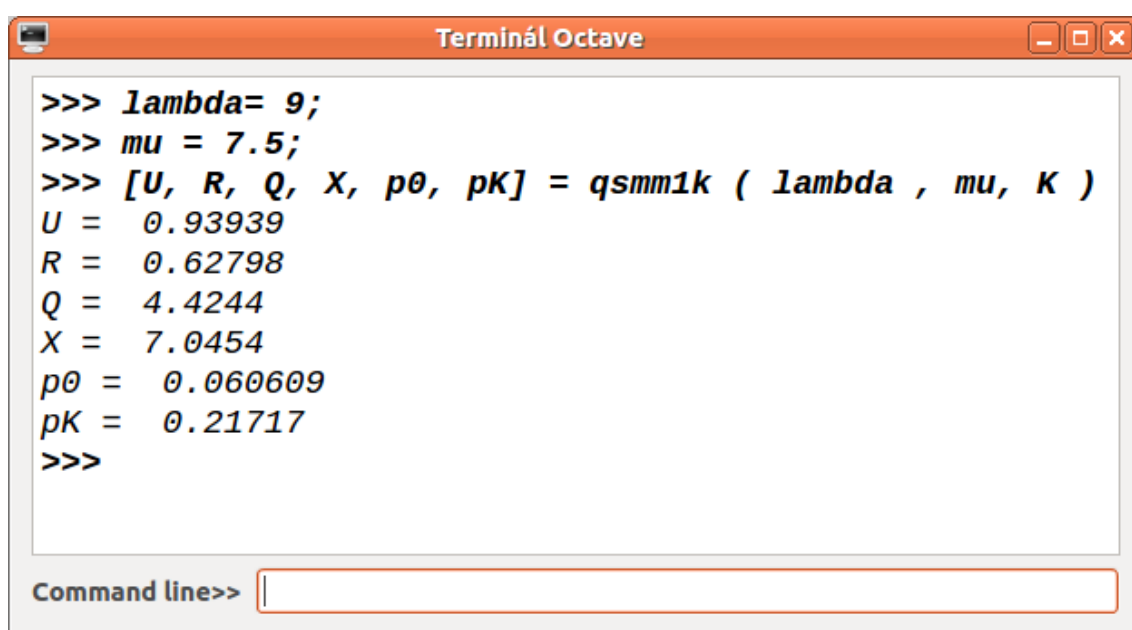
Celková doba strávená v systému:

$$R = \frac{1}{\mu - \lambda} = 0,63 \text{ hodin}$$

Průměrný počet požadavků v systému:

$$Q = \sum_{n=0}^K n * P_n = 4,424 \text{ požadavků}$$

Ukázka v Octave viz **Obr. 7.6**:



```
Terminál Octave
>>> lambda= 9;
>>> mu = 7.5;
>>> [U, R, Q, X, p0, pK] = qsmm1k ( lambda , mu, K )
U = 0.93939
R = 0.62798
Q = 4.4244
X = 7.0454
p0 = 0.060609
pK = 0.21717
>>>
Command line>> |
```

Obrázek 7.6:  $M/M/1K$  - ukázka v Octave

---

## 8 Modelování Markovových řetězců s diskrétním časem

### 8.1 Pravděpodobnost obsazení stavů

Vzhledem k Markovově řetězci s diskrétním časem s množinou stavů  $\{1, 2, \dots, N\}$  označíme  $\pi(n) = (\pi_1(n), \pi_2(n), \dots, \pi_n(n))$  vektor pravděpodobnosti obsazení stavů v kroku  $n$ ,  $n = 0, 1, \dots$ .  $\pi_i(n)$  označuje pravděpodobnost, že je ve stavu  $i$  po  $n$  přechodech.

Je-li dána matice přechodu  $P$  a vektor pravděpodobnosti počátečního stavu obsazení  $\pi(0) = (\pi_1(0), \pi_2(0), \dots, \pi_n(0))$ , pak  $\pi(n)$  můžeme spočítat jako:

$$\pi(n) = \pi(0)P^n \quad (8.1)$$

Za určitých podmínek existuje stacionární pravděpodobnost obsazení stavu  $\pi = \lim_{n \rightarrow +\infty} \pi(n)$ , která je nezávislá na  $\pi(0)$ . Stacionární vektor  $\pi$  je řešením následující soustavy lineárních rovnic:

$$\begin{aligned} \pi P &= \pi \\ \pi \mathbf{1}^T &= 1 \end{aligned}$$

kde  $\mathbf{1}$  je řádkový vektor jedniček a  $(\cdot)^T$  je operátor transpozice.

`p = dtmc (P)`

`p = dtmc (P, n, p0)`

Vypočítá stacionární nebo přechodnou pravděpodobnost obsazenosti stavu pro Markovův řetězec s diskrétním časem.

S jedním argumentem spočítá vektor pravděpodobnosti obsazení stavu  $p(1), \dots, p(N)$  pro Markovův řetězec s diskrétním časem s množinou stavů  $\{1, 2, \dots, N\}$  a maticí přechodu  $P$  o rozměrech  $N \times N$ . S třemi argumenty vypočítá vektor obsazenosti přechodného stavu  $p(1), \dots, p(N)$ , že vektor je ve stavu  $i$  po  $n$  krocích vzhledem k pravděpodobnosti počátečního obsazení  $p0(1), \dots, p0(N)$ .

#### VSTUPNÍ PARAMETRY:

- $P$   $P_{i,j}$  je pravděpodobnost přechodu ze stavu  $i$  do stavu  $j$ .  $P$  musí být neredukovatelná stochastická matice, což znamená, že součet každého řádku musí být 1 ( $\sum_{j=1}^N P_{i,j} = 1$ ) a hodnota matice  $P$  musí být rovna rozměru matice.

- 
- $n$  Počet přechodů, po kterých vypočítá pravděpodobnosti obsazení stavů ( $n = 0, 1, \dots$ ).
  - $p_0$   $p_0(i)$  je pravděpodobnost, že v kroku 0 je systém ve stavu  $i$ .

#### VÝSTUPNÍ PARAMETRY:

- $p$  Pokud je tato funkce volána s jedním parametrem,  $p(i)$  je ustálená pravděpodobnost, že systém je ve stavu  $i$ . Pokud je tato funkce volána se třemi parametry,  $p(i)$  je pravděpodobnost, že systém je ve stavu  $i$  po  $n$  krocích vzhledem k počáteční pravděpodobnosti a počáteční stav je  $i$ .

Příklad:

Uvažujme bludiště s devíti pokoji, jak je znázorněno na následujícím obrázku:

1	2	3
4	5	6
7	8	9

Obrázek 8.1: Příklad s bludištěm

Myš je umístěna v jedné z místností a může bloudit. V každém kroku se myš pohybuje z aktuální místnosti do sousední se stejnou pravděpodobností: je-li v místnosti 1, může se přemístit do místností 2 a 4 s pravděpodobnostmi  $1/2$ . Je-li myš v místnosti 8, může přemístit do místností 5, 7, 9 s pravděpodobnostmi  $1/3$ .

Matice pravděpodobnosti přechodu  $P$  z místnosti  $i$  do místnosti  $j$  je následující:

---


$$\mathbf{P} = \begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

Vektor pravděpodobností obsazenosti stavu může být spočítán použitím následujícího kódu:

```
P = zeros(9,9);
P (1, [2 4]) = 1/2;
P (2, [1 5 3]) = 1/3;
P (3, [2 6]) = 1/2;
P (4, [1 5 7]) = 1/3;
P (5, [2 4 6 8]) = 1/4;
P (6, [3 5 9]) = 1/3;
P (7, [4 8]) = 1/2;
P (8, [7 5 9]) = 1/3;
P (9, [6 8]) = 1/2;
p = dtmc(P);
disp(p)
⇒ 0.083333 0.125000 0.083333 0.125000
    0.166667 0.125000 0.083333 0.125000
    0.083333
```

## 8.2 Birth-Death proces

Vrátí pravděpodobnost přechodu matice P pro diskretní birth-death proces nad množinou stavů  $1, 2, \dots, N$ .  $b(i)$  je pravděpodobnost přechodu ze stavu  $i$  do  $i + 1$  a  $d(i)$  je pravděpodobnost přechodu ze stavu  $i + 1$  do stavu  $i$ , pro  $i = 1, 2, \dots, N - 1$ . Matice P je definována jako:

$$P = \begin{pmatrix} (1 - \lambda_1) & \lambda_1 & & & \\ \mu_1 & (1 - \mu_1 - \lambda_2) & \lambda_2 & & \\ & \mu_2 & (1 - \mu_2 - \lambda_3) & \lambda_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \mu_{N-2} & (1 - \mu_{N-2} - \lambda_{N-1}) & \lambda_{N-1} \\ & & & & \mu_{N-1} & (1 - \mu_{N-1}) \end{pmatrix}$$

kde  $\lambda_i$  a  $\mu_i$  jsou pravděpodobnosti birth a death.

`P = dtmcbd (b, d)`

## 8.3 Očekávaný počet návštěv

Vzhledem k stavu  $N$  diskretního Markovova řetězce s maticí přechodu P a číslem  $n \geq 0$ , označme  $L_i(n)$  jako očekávaný počet návštěv stavu  $i$  během prvních  $n$  přechodů. Vektor  $L(n) = (L_1(n), L_2(n), \dots, L_N(n))$  je definovaný podle **rovnice 8.2**:

$$L(n) = \sum_{i=0}^n \pi(i) = \sum_{i=0}^n \pi(0) P^i \quad (8.2)$$

kde  $\pi(i) = \pi(0)P^i$  je pravděpodobnost obsazení stavu po  $i$  přechodech.

Pokud je P absorbující, to znamená, stochastický proces dosáhne stavu s žádnou možností přechodů s pravděpodobností 1, pak můžeme vypočítat očekávaný počet návštěv před absorpcí L. Nejdříve musíme uspořádat stavy k přepsání matice P jako:

$$P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$$

kde prvních  $t$  stavů jsou přechodné a poslední stavy  $r$  jsou absorbující ( $t + R = N$ ).

---

Matice  $N = (I - Q)^{-1}$  je fundamentální matice;  $N_{i,j}$  je očekávaný počet dob, že proces je v  $j$ -tém přechodném stavu, pokud začal v  $i$ -tém přechodném stavu. Pokud přetvoříme  $N$  k velikosti  $P$  (doplníme chybějící řádky a sloupce nulami), pak dostáváme, že pro absorbující řetězce platí  $L = \pi(0)N$ .

`L = dtmcexps (P, n, p0)`

`L = dtmcexps (P, p0)`

Vypočítá očekávaný počet návštěv do každého stavu během prvních  $n$  přechodů nebo před absorpcí.

#### **VSTUPNÍ PARAMETRY:**

- $P$   $N \times N$  matice přechodu.
- $n$  Počet kroků, během kterých počítáme očekávaný počet návštěv ( $n \geq 0$ ). Když je  $n = 0$ , vrací  $p0$ . Pokud je  $n > 0$ , vrací očekávaný počet návštěv po  $n$  přechodech.
- $p0$  Pravděpodobnost obsazení počátečního stavu.

#### **VÝSTUPNÍ PARAMETRY:**

- $L$  Pokud je funkce volána s dvěma parametry,  $L(i)$  je očekávaný počet návštěv do přechodného stavu  $i$  před absorpcí. Pokud je volána se třemi parametry,  $L(i)$  je očekávaný počet návštěv do stavu  $i$  během prvních  $n$  přechodů vzhledem k počáteční pravděpodobnosti obsazení  $p0$ .



---

## 8.4 Střední čas absorpce

Střední čas absorpce je definovaný jako průměrný počet přechodů, které jsou potřebné k dosažení absorpčního stavu, počínaje přechodným stavem (nebo daným počátečním vektorem pravděpodobnosti obsazení stavu  $\pi(0)$ ).

Nechť  $t_i$  je očekávaný počet přechodů před absorpcí v každém absorpčním stavu začínajícím ve stavu  $i$ . Vektor  $t$  může být spočítán z fundamentální matice  $N$  jako:

$$t = 1N$$

Nechť  $B = [B_{i,j}]$  je matice, kde  $B_{i,j}$  je pravděpodobnost absorpce ve stavu  $j$  začínajícího v přechodném stavu  $i$ . Opět za použití matic  $N$  a  $R$  můžeme psát:

$$B = NR$$

```
[ t N B] = dtmcmmtta (P)
```

```
[ t N B] = dtmcmmtta (P, p0)
```

Vypočítá očekávaný počet kroků do absorpce pro diskrétní Markovův řetězec s  $N \times N$  maticí přechodu  $P$ ; dále spočítá fundamentální matici  $N$  pro  $P$ .

### VSTUPNÍ PARAMETRY:

- $P$   $N \times N$  matice pravděpodobností přechodu.

### VÝSTUPNÍ PARAMETRY:

- $t$  Pokud je funkce volána s jedním parametrem,  $t$  je vektor o velikosti  $N$ , takže  $t(i)$  je očekávaný počet kroků před absorpcí v každém absorpčním stavu počínaje ze stavu  $i$ ; když  $i$  je absorbuje  $t(i) = 0$ . Když je funkce volána s dvěma parametry,  $t$  je skalár reprezentující očekávaný počet kroků před absorpcí počínaje z počátečního stavu pravděpodobnosti obsazení  $p0$ .

- $N$  Pokud je funkce volána s jedním parametrem,  $N$  je  $N \times N$  fundamentální matice pro  $P$ .  $N(i, j)$  je očekávaný počet návštěv do přechodného stavu  $j$  před absorpcí, pokud začal v přechodném stavu  $i$ . Počáteční stav se počítá, pokud je  $i = j$ . Když je funkce volána s dvěma parametry,  $N$  je vektor velikosti  $N$ , takže  $N(j)$  je očekávaný počet návštěv do přechodného stavu  $j$  před absorpcí vzhledem k počátečnímu stavu pravděpodobnosti obsazení  $p_0$ .
- $B$  Pokud je funkce volána s jedním parametrem,  $B$  je  $N \times N$  matice, kde  $B(i, j)$  je pravděpodobnost absorpce ve stavu  $j$  počínaje v přechodném stavu  $i$ ; pokud  $j$  není absorpční,  $B(i, j) = 0$ ; pokud je absorpční,  $B(i, i) = 1$  and  $B(i, j) = 0$  pro všechna  $i \neq j$ . Když je funkce volána s dvěma parametry,  $B$  je vektor velikost  $N$ , kde  $B(j)$  je pravděpodobnost absorpce ve stavu  $j$  vzhledem k počátečnímu stavu pravděpodobnosti obsazení  $p_0$ .

## 8.5 Doba prvního přechodu

Doba prvního přechodu  $M_{i,j}$  je průměrný počet přechodů nutných k první návštěvě stavu  $j$ , vycházíme-li ze stavu  $i$ . Matice  $M$  splňuje následující vlastnost (**rovnice 8.3**):

$$M_{i,j} = 1 + \sum_{k \neq j} P_{i,k} M_{k,j} \quad (8.3)$$

K výpočtu  $M = [M_{i,j}]$  se používá jiná formulace. Nechť  $W$  je  $N \times N$  matice mající každý řádek roven vektoru stacionárního stavu pravděpodobnosti obsazení  $\pi$  matice  $P$  a  $I$  je  $N \times N$  jednotková matice. Definujme  $Z$  podle **rovnice 8.4**:

$$Z = (I - P + W)^{-1} \quad (8.4)$$

Potom dostáváme **rovnici 8.5**:

$$M_{i,j} = \frac{Z_{j,j} - Z_{i,j}}{\pi_j} \quad (8.5)$$

Podle výše uvedené definice  $M_{i,i} = 0$ . Nechť  $M_{i,i}$  je střední doba návratu  $r_i$  pro stav  $i$ , což je průměrný počet přechodů nutných k návratu do stavu  $i$  z jeho původního stavu.  $r_i$  je:

$$r_i = \frac{1}{\pi_i}$$

---

$M = \text{dtmcfpt}(P)$

Vypočítá střední dobu prvního přechodu a střední dobu návratu pro neredukovatelný diskrétní Markovův řetězec.

#### **VSTUPNÍ PARAMETRY:**

- $P$   $P(i, j)$  je pravděpodobnost matice přechodu ze stavu  $i$  do stavu  $j$ .  $P$  musí být neredukovatelná stochastická matice, což znamená, že součet každého řádku musí být roven 1 ( $\sum_{j=1}^N P_{i,j} = 1$ ) a hodnota matice  $P$  musí být rovna dimenzi matice  $P$ .

#### **VÝSTUPNÍ PARAMETRY:**

- $M$  Pro všechna  $i \neq j$ ,  $M(i, j)$  je průměrný počet přechodů než dosáhneme poprvé stavu  $j$ , začínáme-li ve stavu  $i$ .  $M(i, i)$  je střední doba návratu stavu  $i$  a reprezentuje průměrný čas nutný k návratu do stavu  $i$ .

---

## 9 Modelování Markovových řetězců se spojitým časem

### 9.1 Pravděpodobnost obsazení stavů

Podobně jako u diskrétních Markovových řetězců,  $\pi(t) = \pi_1(t), \pi_1(t), \dots, \pi_N(t)$  je pravděpodobnost obsazení stavu v čase  $t$ .  $\pi_i(t)$  je pravděpodobnost, že systém je ve stavu  $i$  v čase  $t \geq 0$ .

Vzhledem k matici intenzit  $Q$  a počátečnímu stavu pravděpodobnosti obsazení  $\pi(0) = (\pi_1(0), \pi_1(0), \dots, \pi_N(0))$ , pravděpodobnost obsazení stavu  $\pi_i(t)$  v čase  $t$  může být spočítána podle **rovnice 9.1**:

$$\pi(t) = \pi(0) \exp(Qt) \quad (9.1)$$

kde  $\exp(Qt)$  je exponenciální matice  $Qt$ . Za určitých podmínek existuje stacionární stav pravděpodobnosti obsazení  $\pi = \lim_{t \rightarrow \infty} \pi(t)$ , který je nezávislý na  $\pi(0)$ .  $\pi$  je výsledkem následující soustavy lineárních rovnic:

$$\pi Q = 0$$

$$\pi 1^T = 1$$

`p = ctmc (Q)`

`p = ctmc (Q, t, p0)`

Vypočítá stacionární nebo přechodný stav pravděpodobnosti pro spojitý Markovův řetězec. S jedním argumentem vypočítá stacionární vektor pravděpodobnosti obsazení stavu  $p(1), \dots, p(N)$  s množinou stavů  $\{1, 2, \dots, N\}$  a  $N \times N$  maticí intenzit  $Q$ . S třemi argumenty vypočítá pravděpodobnost obsazení stavu  $p(1), \dots, p(N)$ , že systém je ve stavu  $i$  v čase  $t$ , vzhledem k počáteční pravděpodobnosti obsazení stavů  $p0(1), \dots, p0(N)$  v čase 0.

#### VSTUPNÍ PARAMETRY:

- $Q$  Matice intenzit.  $Q$  je  $N \times N$  matice, kde  $Q(i,j)$  je hodnota přechodu ze stavu  $i$  do stavu  $j$ , pro  $1 \leq i \neq j \leq N$ .  $Q$  musí splňovat podmínku  $\sum_{j=1}^N Q(i,j) = 0$ .
- $t$  Čas, ve kterém počítáme pravděpodobnost přechodu ( $t \geq 0$ ).
- $p0$   $p0(i)$  je pravděpodobnost, že systém je ve stavu  $i$  v čase 0.

---

## VÝSTUPNÍ PARAMETRY:

- $P$  Pokud je funkce volána s jedním parametrem,  $p(i)$  je ustálená pravděpodobnost, že systém je ve stavu  $i$ , pro  $i = 1, \dots, N$ . Pokud se volá se třemi parametry,  $p(i)$  je pravděpodobnost, že systém je ve stavu  $i$  v čase  $t$  vzhledem k počáteční pravděpodobnost obsazení  $p0(1), \dots, p0(N)$ .

## 9.2 Birth-Death proces

Vrací matici intenzit  $Q$  pro spojitý birth-death proces nad množinou stavů  $1, 2, \dots, N$ .  $b(i)$  je hodnota přechodu ze stavu  $i$  do  $i + 1$  a  $d(i)$  je hodnota přechodu ze stavu  $i + 1$  do stavu  $i$ ,  $i = 1, 2, \dots, N - 1$ .

Matice  $Q$  je definována jako:

$$\begin{pmatrix} -\lambda_1 & \lambda_1 & & & \\ \mu_1 & -(\mu_1 + \lambda_2) & \lambda_2 & & \\ & \mu_2 & -(\mu_2 + \lambda_3) & \lambda_3 & \\ & & \ddots & \ddots & \ddots \\ & & & \mu_{N-2} & -(\mu_{N-2} + \lambda_{N-1}) & \lambda_{N-1} \\ & & & & \mu_{N-1} & -\mu_{N-1} \end{pmatrix}$$

kde  $\lambda_i$  je hodnota birth a  $\mu_i$  je hodnota death.

$Q = \text{ctmcbd}(b, d)$

---

### 9.3 Střední čas absorpce

Vypočítá střední dobu absorpce spojitého Markovova řetězce popsaného maticí intenzit  $Q$  a počáteční pravděpodobností obsazení  $p$ . Pokud zde není žádný absorbující stav, funkce vrátí chybu.

`t = ctmcmmta (Q, p)`

#### VSTUPNÍ PARAMETRY:

- $Q$   $N \times N$  matice intenzit.  $Q(i,j)$  je hodnota přechodu ze stavu  $i$  do stavu  $j$ ,  $i \neq j$ . Matice  $Q$  musí splňovat podmínku  $\sum_{j=1}^N Q_{i,j} = 0$ .
- $p$   $p(i)$  je pravděpodobnost, že systém je ve stavu  $i$  v čase 0, pro každé  $i = 1, \dots, N$ .

#### VÝSTUPNÍ PARAMETRY:

- $t$  Střední čas absorpce.

### 9.4 Doba prvního přechodu

`m = ctmcfpt (Q, i, j)`

Vypočítá dobu prvního přechodu pro neredukovatelný spojitý Markovův řetězec.

#### VSTUPNÍ PARAMETRY:

- $Q$  Matice intenzit.  $Q$  je  $N \times N$  matice, kde  $Q(i,j)$  je hodnota přechodu ze stavu  $i$  do stavu  $j$ , pro  $1 \leq i \neq j \leq N$ . Hodnoty přechodů nesmí být záporné a  $\sum_{j=1}^N Q_{i,j} = 0$ .
- $i$  Počáteční stav.
- $j$  Konečný stav.

---

## VÝSTUPNÍ PARAMETRY:

$m$   $m$  je průměrná doba, než navštívíme poprvé stav  $j$ , vycházíme-li ze stavu  $i$ .

Příklad:

Je dána matice intenzit:

$$Q = \begin{pmatrix} -1.0 & 0.9 & 0.1 \\ 0.1 & -1.0 & 0.9 \\ 0.9 & 0.1 & -1.0 \end{pmatrix}$$

Určíme, jaká je doba prvního přechodu ze stavu 1 do stavu 3.

```
Q = [ -1.0  0.9  0.1; 0.1 -1.0  0.9; 0.9  0.1 -1.0 ];
```

```
m = ctmcftp(Q,1,3)
```

```
m = 2.0879
```

Doba prvního přechodu ze stavu 1 do stavu 3 je  $m = 2.0879$  s.

---

## 10 Návrh laboratorního cvičení

1. Nainstalujte balíček Queueing do programu Octave.
2. Pomocí programu Octave a jeho rozšiřujícího balíčku vypočítejte následující příklady:

### a) M/M/1 systém

Mějme letištní dráhu pouze pro přílety. Přistávající letadla se připojují do fronty pro přistání. Dále mějme exponenciální rozložení doby obsluhy s intenzitou  $\mu = 27$  přiletů/hod. Letadla přilétají s intenzitou  $\lambda = 20$  přiletů/hod. Určete využití systému, čas, který letadlo stráví ve frontě a v obsluze (tzn. celkový čas strávený v systému), průměrný počet požadavků v systému a pravděpodobnost, že v systému není žádné letadlo.

### Řešení:

```
lambda = 20;  
mu = 27;  
[U, R, Q, X, p0] = qsmml(lambda, mu)
```

### b) Příklad Erlang B

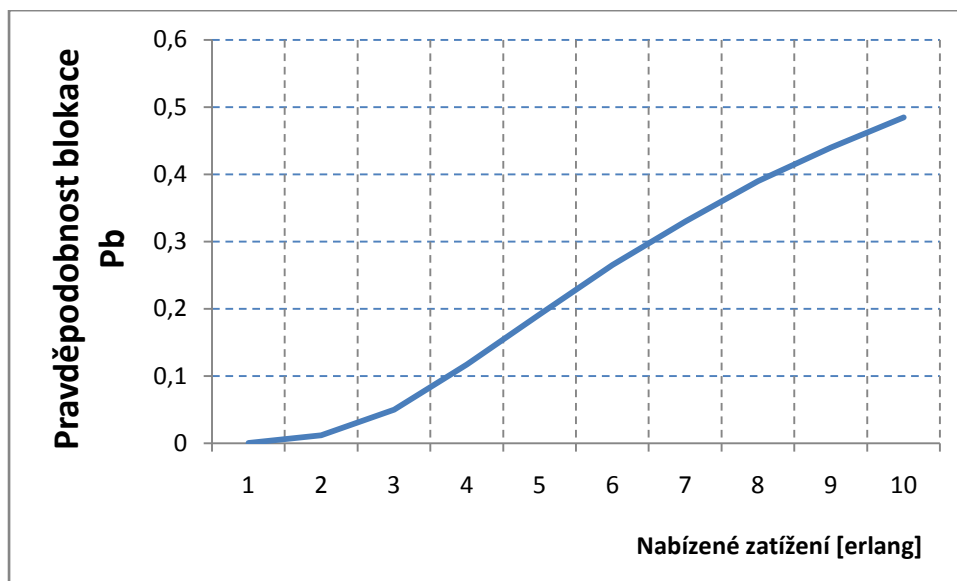
Pomocí Erlangovy první rovnice určete pravděpodobnost blokace požadavku, jestliže máme zadáno nabízené zatížení  $A = 3$  erlang a systém obsahuje 6 obsluhových linek. Výsledek ověřte v programu Octave pomocí balíčku Queueing. Postupně proveďte simulaci pro  $A = 1, 2, \dots, 10$  a sestrojte graf závislosti pravděpodobnosti blokace na nabízeném zatížení  $A$ .

### Řešení:

$$E_B(A, m) = \frac{A^m}{m!} \left( \sum_{k=0}^m \frac{A^k}{k!} \right)^{-1} = \frac{3^6}{6!} \left( \frac{3^0}{0!} + \frac{3^1}{1!} + \frac{3^2}{2!} + \frac{3^3}{3!} + \frac{3^4}{4!} + \frac{3^5}{5!} + \frac{3^6}{6!} \right)^{-1} = 0,522 = 5,22 \%$$

```
A = 3;  
m = 6;  
B = erlangb (A, m)  
B = 0.052
```





Obrázek 10.1: Graf závislosti pravděpodobnosti blokace na nabízeném zatížení

3. Pro vyvolání nápovědy k dané funkci použijte příkaz `help`, například pro nápovědu k funkci pro modelaci M/M/1 systému použijte příkaz: `help qsmmm1`.

---

## Závěr

V práci byla uvedena teorie základních obsluhových systémů s popisem jednotlivých částí systému, jejich zjednodušená i rozšířená klasifikace s popisem jednotlivých označení. Byly popsány modely OS a Markovovy řetězce včetně základních rovnic popisujících základní vztahy. Postupně jsou popsány jednotlivé definice obsluhových systémů a jejich modelování v tomto programu. Každá kapitola vztahující se k modelování obsluhových systému a Markovových řetězců popisuje vstupní a výstupní parametry k dané funkci. Octave Queueing Package je volně dostupný v různých modifikacích pro operační systémy a poskytuje řadu funkcí pro modelování jak obsluhových systému, tak Markovových řetězců a umožňuje jejich zobrazení pomocí grafů. Práce slouží jako návod pro program GNU Octave a jeho rozšiřující balíček GNU Octave, v práci bylo uvedeno několik jednoduchých příkladů pro modelování těchto systémů s vysvětlujícím popisem řešení, grafy a obrázky. Poslední kapitola je určena pro odborný předmět jako námět laboratorního cvičení se základními příklady, kde si student vyzkouší jednotlivé funkce a jejich modelování. Ke každému příkladu bylo uvedeno i řešení. Práce s program GNU Octave není nijak složitá, program umožňuje vyvolání nápovědy ke každé funkci.

---

## Seznam použité literatury

- [1] ZÍTEK, František. Ztracený čas: (elementy teorie hromadné obsluhy). Vyd. 1. Praha: Academia, 1969, 179 s.
- [2] KŘÍŽOVSKÝ, František. Provozní zatížení v telekomunikacích [online]. Praha, 1990 [cit. 2014-01-09]. skripta.
- [3] MARZOLLA, Moreno. A Queueing Package for GNU Octave. [Http://www.moreno.marzolla.name/](http://www.moreno.marzolla.name/) [online]. Bologna, 2012 [cit. 2014-01-09]. Dostupné z: <http://www.moreno.marzolla.name/software/queueing/>
- [4] BOCHAROV, P. Queueing theory. Boston: VSP, 2004, xiv, 446 p. ISBN 90-676-4398-X.
- [5] HAGHIGHI, Aliakbar Montazer a D MISHEV. Queuing models in industry and business. New York: Nova Science Publishers, c2008, xii, 386 p. ISBN 16-045-6189-0.
- [6] GROSS, Donald a Carl M HARRIS. Fundamentals of queueing theory. 3rd ed. New York: John Wiley, c1998, xi, 439 s. ISBN 04-711-7083-6.
- [7] KLEINROCK, Leonard. Queuein gsystems: Volume 1: Theory. New York: John Wiley, c1975, xviii, 417 s. ISBN 04-714-9110-1.
- [8] MorenoMarzolla, The qnetworks Toolbox: A Software Package for Queueing Networks Analysis. Khalid Al-Begain, Dieter Fiems and William J. Knotten- belt, Editors, Proceedings 17th International Conference on Analytical and Stochastic Modeling Techniques and Applications (ASMTA 2010) Cardiff, UK, June 14–16, 2010, volume 6148 of Lecture Notes in Computer Science, Springer, pp. 102–116, ISBN 978-3-642-13567-5
- [9] Takács, L. Introduction to the theory of queues. Oxford University Press, New York, 1962.